

TUGAS AKHIR - KI141502

DETEKSI GEMPA BERDASARKAN DATA TWITTER MENGUNAKAN DECISION TREE, RANDOM FOREST DAN SVM

RENDRA DWI LINGGA P.
NRP 5111100008

Dosen Pembimbing
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.
Diana Purwitasari, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

DETEKSI GEMPA BERDASARKAN DATA TWITTER MENGUNAKAN DECISION TREE, RANDOM FOREST DAN SVM

RENDRA DWI LINGGA P.
NRP 5111100008

Dosen Pembimbing
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.
Diana Purwitasari, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

EARTHQUAKE DETECTION OF TWITTER DATA USING DECISION TREE, RANDOM FOREST AND SVM

Rendra Dwi Lingga P
NRP 5111100008

Advisor
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.
Diana Purwitasari, S.Kom., M.Sc.

INFORMATICS DEPARTMENT
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

DETEKSI GEMPA BERDASARKAN DATA TWITTER MENGGUNAKAN DECISION TREE, RANDOM FOREST, DAN SVM

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat Memperoleh
Gelar Sarjana Komputer pada Rumpun Mata Kuliah
Komputasi Cerdas dan Visi
Program Studi S-1 Jurusan Teknik Informatika Fakultas
Teknologi Informasi Institut Teknologi Sepuluh Nopember

Oleh :

RENDRA DWI LINGGA P.

NRP : 5111100008

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.

NIP: 197512202001122002

(Pembimbing 1)

Diana Purwitasari, S.Kom., M.Sc.

NIP: 197804102003122001

(Pembimbing 2)

**SURABAYA
JANUARI 2017**

[Halaman ini sengaja dikosongkan]

DETEKSI GEMPA BERDASARKAN DATA TWITTER MENGUNAKAN DECISION TREE, RANDOM FOREST DAN SVM

Nama Mahasiswa : Rendra Dwi Lingga P
NRP : 5111100008
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing I : Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.
Dosen Pembimbing II : Diana Purwitasari, S.Kom., M.Sc.

ABSTRAK

Twitter merupakan salah satu media sosial yang cukup populer saat ini. Pengguna aktif Twitter mencapai kurang lebih 400 juta orang. Fitur utama yang paling penting dari Twitter yaitu layanan yang bersifat real-time dimana pengguna dapat menuliskan catatan singkat tentang apa yang terjadi secara langsung. Sebagai contoh, ketika terjadi bencana alam(gempa bumi) di suatu tempat, banyak pengguna aktif twitter menulis informasi berupa (tweet) tentang gempa bumi yang sedang berlangsung melalui Twitter. Hal ini memungkinkan dibuatnya sebuah metode yang mendeteksi terjadinya gempa atau tidak dengan melakukan observasi melalui tweet yang ada.

Dalam tugas akhir ini dibuat sebuah metode klasifikasi untuk membedakan antara tweet yang mengandung informasi gempa yang sesungguhnya (gempa positif) dan tweet yang mengandung informasi gempa namun memiliki arti lain (gempa negatif).

Setelah dilakukan klasifikasi menggunakan Decision Tree, Random Forest dan Support Vector Machine (SVM). Hasil yang didapat memberikan nilai akurasi Support Vector Machine (SVM) secara keseluruhan lebih baik daripada Decision Tree dan Random Forest dengan persentase gempa yang dideteksi oleh sistem (Recall) didapatkan nilai 86.3%.dengan precision sebesar 88.7%. Namun jika dilihat dari terdeteksinya gempa oleh sistem

tanpa dirata-rata, Random Forest memiliki persentase recall sebesar 96.7% lebih baik daripada Decision Tree dan Random Forest.

Kata kunci: Twitter, deteksi kejadian, gempa.

EARTHQUAKE DETECTION OF TWITTER DATA USING DECISION TREE, RANDOM FOREST AND SVM

Name : Rendra Dwi Lingga P
NRP : 5111100008
Major : Informatics Department, FTIf-ITS
Advisor I : Dr.Eng. Chastine Fatichah, S.Kom.,
M.Kom.
Advisor II : Diana Purwitasari, S.Kom., M.Sc.

ABSTRACT

Twitter is one of media social that very popular. On the record twitter active users are 400 billion. The main and the most important feature is real-time quotes, which is user could share short quotes at real time. For example is about earthquake at somewhere, user would be reflect "tweet" regarding that earthquake. There for it is important to have detection and observation method related with real-time quote (example:earthquake).

In this thesis is explained about clarifications method to make differences between real/actual information/tweet (read : positive earthquake) and information/tweet that has other explanation (read : negative earthquake)

The clarification methods are using Decision Tree, Random Forest, and Support Vector Machine (SVM). The result is better value of accurate Support Vector Machine (SVM) than Decision Tree and Random Forest. The percentage of detection system (recall) is 86.3%, but when using precision is increase become 88.7%. But, if we compare earthquake disaster from system that no need to average, Randon Forest has recall percentage 96.7%, that value is better than Decision Tree and Random Forest.

Keywords: *Twitter, event detection, earthquake.*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Segala puji bagi Allah SWT, yang telah melimpahkan rahmat-Nya kepada penulis, sehingga Tugas Akhir yang berjudul ***“DETEKSI GEMPA BERDASARKAN DATA TWITTER MENGGUNAKAN DECISION TREE, RANDOM FOREST DAN SVM”*** ini dapat selesai tepat waktu.

Dalam pengerjaan Tugas Akhir ini penulis mendapatkan bantuan dan juga motivasi dari berbagai pihak. Di kesempatan ini penulis ingin memberikan terima kasih sebesar-besarnya kepada:

1. Orang tua dan keluarga penulis yang senantiasa memberikan doa, motivasi, dan juga dukungan yang besar kepada penulis agar selesai dalam proses pengerjaan Tugas Akhir ini.
2. Ibu Dr.Eng. Chastine Fatichah, S.Kom., M.Kom. dan Ibu Diana Purwitasari, S.Kom., M.Sc. sebagai dosen pembimbing yang telah memberikan ilmu, waktu, motivasi dan kesabarannya kepada penulis dalam proses pengerjaan Tugas Akhir ini.
3. Pak Prof.Ir. Supeno Djanali, M.Sc., Ph.D sebagai dosen wali yang telah memberikan bimbingan selama masa perkuliahan.
4. Pak Dr.Eng. Radityo Anggoro, S.Kom., M.Sc. sebagai dosen Koordinator Tugas Akhir yang telah memberikan bimbingan dan nasihat selama masa pengerjaan Tugas Akhir ini.
5. Teman-teman seangkatan 2011 yang telah memberikan banyak dukungan dalam pengerjaan Tugas Akhir ini.
6. Rosi dan Rizka yang telah memberikan diskusi dan saran kepada penulis.
7. Serta semua pihak yang telah memberikan bantuan dalam pengerjaan Tugas Akhir ini.

Penulis menyadari bahwa dalam pengerjaan Tugas Akhir ini terdapat banyak kekurangan. Oleh karena itu, kritik serta saran

akan sangat bermanfaat untuk penyempurnaan Tugas Akhir secara lebih lanjut. Akhir kata, penulis meminta maaf bila penulisan laporan Tugas Akhir ini masih banyak kekurangan. Semoga hasil dari Tugas Akhir ini dapat memberikan manfaat kepada pembaca.

Surabaya, Januari 2017

Rendra Dwi Lingga P.

DAFTAR ISI

Abstrak	ix
Abstract	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xviii
DAFTAR TABEL	xx
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Permasalahan.....	2
1.3. Batasan Permasalahan	2
1.4. Tujuan.....	2
1.5. Metodologi	2
1.6. Sistematika Penulisan.....	4
BAB II DASAR TEORI.....	6
2.1. Twitter	6
2.2. Gempa Bumi.....	7
2.3. Support Vector Machine	7
2.4. Random Forest	8
2.5. Decision Tree	9
2.6. Cross-validation.....	10
BAB III ANALISIS DAN	13
PERANCANGAN SISTEM.....	13
3.1. Perancangan Data.....	13
3.2. Perancangan Sistem.....	14
3.2.1. Ekstraksi Fitur	17
3.2.2. Perbandingan Klasifikasi Decision Tree, Random Forest dan SVM	18
BAB IV IMPLEMENTASI.....	20
4.1. Lingkungan Implementasi.....	20
4.2. Implementasi Modul.....	20
4.2.1. Implementasi Modul Conn.php.....	20
4.2.2. Implementasi Modul Index.php.....	21
4.2.3. Implementasi Modul Ganti_Kata_Class.php.....	24

4.2.4.	Implementasi Modul Export_Matriks2.php.....	26
4.2.5.	Implementasi Modul Matriks2.php	26
4.2.6.	Tabel MySql	28
4.3.	WEKA	32
BAB V PENGUJIAN DAN EVALUASI		36
5.1.	Lingkungan Uji Coba	36
5.2.	Data Uji Coba.....	36
5.3.	Skenario Uji Coba	36
5.3.1.	Skenario Uji Coba Akurasi Klasifikasi Tweet dengan metode Decision Tree, Random Forest dan SVM ..	37
5.4.	Hasil Uji Coba	38
5.4.1.	Hasil Uji Coba Akurasi Metode Klasifikasi <i>Tweet</i> menggunakan Decision Tree	38
5.4.2.	Hasil Uji Coba Akurasi Metode Klasifikasi <i>Tweet</i> menggunakan Random Forest	39
5.4.3.	Hasil Uji Coba Akurasi Metode Klasifikasi <i>Tweet</i> menggunakan SVM	40
5.4.4.	Hasil Perbandingan Persentasi Weighted Avg dari Decision Tree, Random Forest dan SVM.....	41
5.4.5.	Hasil Uji Coba Perbandingan Persentase Recall yang Dideteksi oleh Sistem	42
BAB VI KESIMPULAN DAN SARAN		44
6.1.	Kesimpulan.....	44
6.2.	Saran	44
DAFTAR PUSTAKA.....		46
BIODATA PENULIS.....		48

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi Twitter.....	6
Gambar 2.2 Ilustrasi SVM.....	7
Gambar 2.3 Ilustrasi Random Forest.....	8
Gambar 2.4 Ilustrasi Decision Tree.....	10
Gambar 3.1 Arsitektur Umum Sistem.....	14
Gambar 4.1 Kode program conn.php	21
Gambar 4.2 Kode program index.php	23
Gambar 4.3 Halaman Antarmuka.....	24
Gambar 4.4 Kode program export_matriks2.php	26
Gambar 4.5 Kode program evaluasi model di tweet_classifier...	26
Gambar 4.6 Kode program matriks2.php	28
Gambar 4.7 Tabel MySql	29
Gambar 4.8 Tabel Stopword.....	29
Gambar 4.9 Tabel Twitter	30
Gambar 4.10 Tabel twitter_class.....	30
Gambar 4.11 Tabel term_twitter	31
Gambar 4.12 Tabel term_twitter_class.....	31
Gambar 4.13 Halaman Awal WEKA	32
Gambar 4.14 Open File	32
Gambar 4.15 Preprocess.....	33
Gambar 4.16 Pilihan Klasifikasi.....	33
Gambar 4.17 Klasifikasi Decision Tree	34
Gambar 4.18 Klasifikasi SVM	34
Gambar 4.19 Random Forest.....	34

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Eksperimen Cross-Validation.....	11
Tabel 3.1 Tokenisasi.....	15
Tabel 3.2 Stopword Removal	15
Tabel 3.3 Hilangkan Angka.....	16
Tabel 3.4 Normalisasi.....	16
Tabel 3.5 Hilangkan Karakter	17
Tabel 3.6 Perhitungan Term Frequency	17
Tabel 4.1 Spesifikasi perangkat keras	20
Tabel 5.1 Confusion Matrix	37
Tabel 5.2 Hasil Uji Coba Akurasi Metode Klasifikasi Decision.....	38
Tabel 5.3 Hasil Uji Coba Akurasi Metode Klasifikasi Random	39
Tabel 5.4 Hasil Uji Coba Akurasi Metode Klasifikasi SVM 3- fold, 5-fold dan 10-fold Cross-Validation	40
Tabel 5.5 Perbandingan <i>weighted avg</i>	41
Tabel 5.6 Nilai <i>Recall</i> dari Sistem dalam Mendeteksi Gempa	42

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

1.1. Latar Belakang

Twitter merupakan salah satu media sosial yang cukup terkenal. Pada tahun 2016, jumlah pengguna aktif Twitter mencapai 400 juta orang. Fitur utama dari Twitter yaitu *tweet* dimana pengguna dapat menuliskan catatan singkat tentang apa yang terjadi di lingkungan sekitarnya secara langsung. Twitter bersifat *real-time* sehingga membuat Twitter menjadi media sosial yang menarik digunakan untuk berbagai metode *event detection*, salah satunya bencana alam gempa bumi [1].

Deteksi tentang adanya gempa adalah hal yang cukup krusial, bahkan bisa menyelamatkan nyawa. Bila seseorang mengetahui akan ada gempa beberapa detik saja sebelum gempa terasa, maka waktu itu bisa digunakan untuk hal penting seperti mematikan saluran gas.

Data *tweet* yang didapatkan dari Twitter, bisa menjadi dasar untuk mendeteksi terjadinya gempa bumi. Akibat data *tweet* yang banyak mengandung kata tidak baku, maka perlu dibuat sebuah model klasifikasi untuk menentukan apakah *tweet* benar-benar memberi informasi tentang adanya gempa saat ini.

Dari permasalahan yang telah dipaparkan di atas, dalam tugas akhir ini akan diimplementasikan sebuah metode klasifikasi deteksi gempa di wilayah Indonesia berdasarkan informasi yang terdapat pada Twitter. Dengan adanya metode ini diharapkan dapat memberikan informasi dini yang akurat terkait ada atau tidaknya gempa.

1.2. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana melakukan *preprocessing* terhadap data Twitter yang didapat?
2. Bagaimana mengimplementasikan metode klasifikasi yang mampu menentukan *tweet* yang memberikan informasi tentang adanya gempa saat ini?
3. Bagaimana mengevaluasi kinerja metode deteksi gempa?

1.3. Batasan Permasalahan

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain:

1. Dataset *tweet* yang digunakan adalah *tweet* dengan Bahasa Indonesia.
2. Diasumsikan tidak terjadi dua atau lebih gempa secara bersamaan.
3. Gempa yang dideteksi adalah gempa dirasakan yang terjadi di wilayah Indonesia.

1.4. Tujuan

Tujuan dari pembuatan tugas akhir ini adalah membuat metode yang mampu mendeteksi terjadinya gempa atau tidak gempa di Indonesia dan berdasarkan data Twitter.

1.5. Metodologi

Tahap yang dilakukan untuk menyelesaikan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan Proposal Tugas Akhir

Penyusunan proposal ini merupakan tahap awal dalam pengerjaan Tugas Akhir. Proposal tugas akhir ini terdiri atas latar belakang, rumusan masalah, batasan masalah, serta tujuan dan manfaat dari tugas akhir. Selain itu dijelaskan juga

tinjauan pustaka yang digunakan untuk referensi dalam pembuatan tugas akhir.

2. Studi Literatur

Pada tahap studi literatur ini, akan dipelajari berbagai referensi yang dibutuhkan dalam pembuatan tugas akhir, seperti pemrograman PHP, algoritma *Decision Tree*, algoritma *Random Forest* dan algoritma *Support Vector Machine* dan penggunaan aplikasi WEKA.

3. Implementasi

Pada tahapan ini dibangun perangkat lunak sesuai dengan rancangan yang diajukan pada proposal. Pembangunan program diimplementasikan sesuai dengan konsep yang telah didapatkan saat studi literatur.

4. Pengujian dan Evaluasi

Ada 3 evaluasi yang akan diterapkan pada tugas akhir ini, yaitu:

1. Evaluasi akurasi klasifikasi tweet menggunakan Decision Tree

Evaluasi ini akan dilakukan dengan melihat akurasi (%) dari metode decision tree. Berapa banyak tweet yang terklasifikasi sesuai dengan kelas seharusnya. Data pengujian ini diambil dari data historis tweet yang telah ada.

2. Evaluasi akurasi klasifikasi tweet menggunakan Random Forest

Evaluasi ini akan dilakukan dengan melihat akurasi (%) dari metode random forest. Berapa banyak tweet yang terklasifikasi sesuai dengan kelas seharusnya. Data pengujian ini diambil dari data historis tweet yang telah ada.

3. Evaluasi akurasi metode klasifikasi menggunakan SVM

Evaluasi ini akan dilakukan dengan melihat akurasi (%) dari metode SVM. Berapa banyak tweet yang terklasifikasi sesuai dengan kelas seharusnya. Data pengujian ini diambil dari data historis tweet yang telah ada.

4. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari metode klasifikasi yang telah dibuat.

1.6. Sistematika Penulisan

Buku Tugas Akhir ini disusun dengan sistematika sebagai berikut:

1. Bab I Pendahuluan

Bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, dan sistematika laporan Tugas Akhir.

2. Bab II Tinjauan Pustaka

Bab ini berisi tentang teori yang digunakan dan diimplementasikan pada Tugas Akhir. Teori-teori tersebut mencakup konsep dasar serta algoritma yang digunakan untuk menyelesaikan permasalahan.

3. Bab III Perancangan

Bab ini berisi gambaran tugas akhir secara umum. Pada bab ini dijelaskan tahapan-tahapan yang dijalankan oleh aplikasi.

4. Bab IV Implementasi

Bab ini berisi hasil implementasi rancangan aplikasi berupa kode program dalam bahasa pemrograman PHP dan penggunaan aplikasi WEKA.

5. Bab V Uji Coba dan Evaluasi

Bab ini berisi hasil evaluasi tugas akhir dengan menggunakan program yang dibangun. Juga disertakan analisis dari hasil evaluasi aplikasi,

6. Bab VI Kesimpulan dan Saran

Bab ini merupakan penjelasan berupa hasil akhir yang dapat ditarik dari keseluruhan proses dan percobaan Tugas Akhir. Selain itu di bab ini juga terdapat saran-saran yang berisi hal-hal yang dapat diperbaiki dan dikembangkan.

[Halaman ini sengaja dikosongkan]

BAB II DASAR TEORI

Pada bab ini akan dibahas mengenai dasar teori yang menjadi dasar pembuatan tugas akhir ini.

2.1. Twitter

Twitter adalah sosial media yang memberikan layanan pada penggunanya untuk berbagi hal-hal yang sedang terjadi saat itu juga. Twitter menanyakan satu pertanyaan, “*What’s happening?*” (Apa yang sedang terjadi?), kepada penggunanya. Jawaban yang diberikan oleh pengguna tidak boleh lebih dari 140 karakter [1].

Salah satu aspek penting dari Twitter adalah karakteristiknya yang bersifat *real-time*. Sebagai contoh, saat terjadi gempa, banyak pengguna mengirim jawaban ke Twitter (*tweets*) yang berhubungan dengan gempa. Hal ini memungkinkan dilakukan deteksi terhadap gempa dengan menginspeksi berbagai *tweet* yang masuk ke Twitter.



Gambar 2.1 Ilustrasi Twitter

2.2. Gempa Bumi

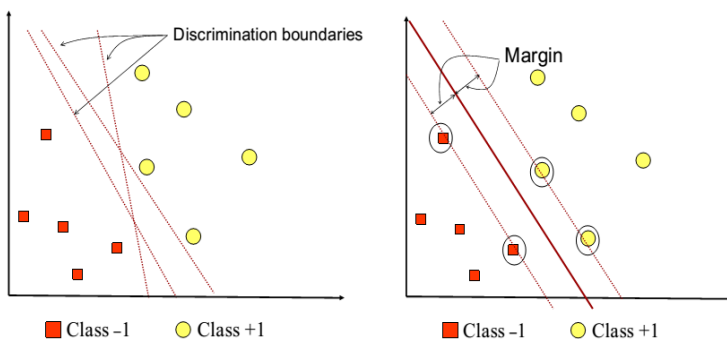
Gempa bumi adalah getaran di permukaan bumi yang diakibatkan oleh pelepasan energi dari dalam secara tiba-tiba sehingga menciptakan gelombang seismik. Gempa bumi di Indonesia biasa diakibatkan oleh pertemuan antara plat tektonik [2].

Kekuatan dari gempa ditentukan dari amplitude getaran gempa yang diukur dari *seismograph*, serta jarak *seismograph* tersebut dari pusat gempa. Kekuatan dari gempa biasa diukur dalam skala Magnitude, yang memperlihatkan seberapa besar energi yang dikeluarkan oleh gempa.

Indonesia merupakan salah satu negara yang cukup sering terkena gempa bumi, hal ini diakibatkan oleh letak geografis dari Indonesia. Gempa yang memakan banyak korban di Indonesia terjadi pada tahun 2006 di pulau Jawa serta pada tahun 2004 di Aceh yang juga disertai oleh gelombang tsunami [3].

2.3. Support Vector Machine

SVM atau *Support Vector Machine* adalah sebuah metode pembelajaran *supervised* yang menganalisis data dan mengidentifikasi pola. SVM biasa digunakan dalam klasifikasi data menjadi kelas tertentu [4].



Gambar 2.2 Ilustrasi SVM

SVM bekerja dengan cara memetakan data *training* yang telah memiliki label menjadi titik-titik di ruang. Titik-titik tersebut dipetakan sedemikian rupa sehingga jarak dari titik terdekat antara masing-masing kelas *training data* menjadi sejauh mungkin.

SVM pada dasarnya adalah sebuah model klasifikasi linear, namun bisa digunakan untuk melakukan klasifikasi non-linear dengan *kernel trick*. *Kernel trick* dalam SVM adalah cara untuk melakukan klasifikasi data non-linear dengan cara mentransformasikan ruang data asli menjadi ruang data dengan dimensi yang lebih tinggi. *Kernel* yang biasa digunakan dalam SVM antara lain *Gaussian Radial Basis Function* (RBF), Polynomial, dan Hyperbolic.

2.4. Random Forest

Random Forest adalah sebuah metode bisa yang digunakan untuk klasifikasi, regresi, ataupun tujuan lainnya. *Random Forest* bekerja dengan cara membangun lebih dari 1 *Decision Tree* secara random saat *training*. Hasil yang diberikan oleh *Random Forest* untuk klasifikasi adalah modus dari *decision tree-decision tree* nya. Sementara nilai yang diberikan untuk regresi adalah mean [4].

Dengan membuat banyak *Decision Tree* secara random, maka sebenarnya banyak dari pohon-pohon yang dibuat oleh metode *Random Forest* menjadi kurang berguna. Namun



Gambar 2.3 Ilustrasi Random Forest

Random Forest mampu menjadi sebuah metode klasifikasi yang

cukup baik, karena beberapa *Decision Tree* yang ikut dibuat saat konstruksi, ternyata memiliki kemampuan prediksi yang baik. Saat dilakukan pemilihan untuk menentukan klasifikasi secara keseluruhan, pohon-pohon yang buruk akan membuat prediksi yang acak dan saling bertentangan, sehingga jawaban dari beberapa decision tree yang merupakan prediktor yang baik akan muncul sebagai jawaban.

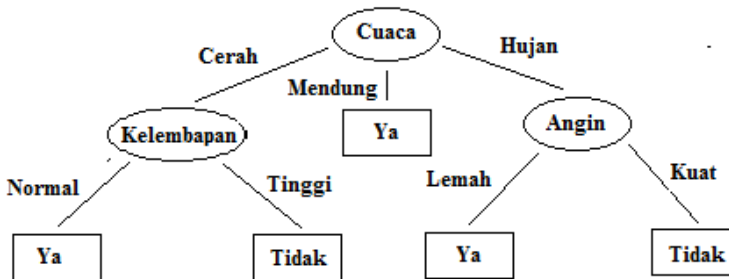
Random Forest pertama kali dipublikasikan secara resmi oleh Leo Breiman pada tahun 2001 [4]. *Random Forest* dikembangkan untuk memperbaiki metode-metode *Decision Tree* yang rawan *overfitting*.

Dalam perkembangannya *Random Forest* menjadi salah satu metode yang populer di bidang *machine learning*. Hal ini diakibatkan oleh mudahnya penggunaan *Random Forest*, yang mampu mencapai akurasi tinggi tanpa perlu melakukan banyak *parameter tuning* [5].

2.5. Decision Tree

Decision Tree adalah sebuah metode klasifikasi yang dibangun untuk mendapatkan sebuah kesimpulan dari sejumlah data. Penarikan kesimpulan dibuat dalam bentuk pohon, dimana nantinya hasil kesimpulan berbentuk hierarki pohon yaitu dari akar, batang dan daun yang merepresentasikan hasil keputusan yang dibuat.

Sebuah *node* keputusan (misalnya, Cuaca) memiliki dua cabang atau lebih (misalnya, cerah, mendung dan hujan). *Node* daun (misalnya, *Play*) merupakan klasifikasi atau keputusan. *Node* keputusan paling atas di pohon adalah yang sesuai dengan prediktor terbaik disebut *node* akar [6].



Gambar 2.4 Ilustrasi Decision Tree

Penerapan *Decision Tree* dapat dilihat pada Gambar 2.4. Pada gambar tersebut dalam dilihat bawah *Decision Tree* memiliki 2 macam *node* yaitu *node* keputusan (Cuaca) dan *node* daun (*Play*=Ya atau *Play*=Tidak).

2.6. Cross-validation

Cross-validation merupakan salah satu teknik untuk menilai keakuratan sebuah model yang dibangun berdasarkan dataset tertentu. Pembuatan model biasanya bertujuan untuk melakukan prediksi maupun klasifikasi terhadap suatu data baru yang boleh jadi belum pernah muncul di dalam dataset. Data yang digunakan dalam proses pembangunan model disebut data training., sedangkan data yang digunakan untuk menilai model disebut sebagai data test.

Salah satu metode cross-validation yang populer dalah K-Fold Cross-validation. Dalam teknik ini dataset dibagi menjadi sejumlah K-buah partisi secara acak. Kemudian dilakukakn sejumlah K-kali eksperimen, dimana masing-masing eksperimen menggunakan data partisi ke-K sebagai data testing dang memanfaatkan sisa partisi lainnya sebagai data training. Sebagai

gambaran, jika kita melakukan 5-Fold Cross-validation maka desain data eksperimennya seperti Tabel 2.1:

Dataset : K1, K2, K3, K4, K5

Data Eksperimen :

Tabel 2.1 Eksperimen Cross-Validation

Ekperimen Ke	Data Latih	Data Test
1	K2,K3,K4,K5	K1
2	K1,K3,K4,K5	K2
3	K1,K2,K4,K5	K3
4	K1,K2,K3,K5	K4
5	K1,K2,K3,K4	K5

[Halaman ini sengaja dikosongkan]

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini akan dibahas perancangan data, sistem, dan metode yang digunakan. Metode-metode yang digunakan adalah klasifikasi tweet menggunakan metode *Decision Tree*, *Random Forest* dan SVM.

3.1. Perancangan Data

Pada bagian ini dijelaskan perancangan data yang akan digunakan dalam uji coba.

Perancangan Data Masukan

Data masukan dari sistem adalah data *tweet* yang didapatkan melalui Twitter. Data *tweet* yang didapatkan memiliki atribut antara lain: teks/isi dari *tweet*.

Data yang digunakan dalam tugas akhir ini adalah data *tweet* yang mengandung kata gempa dari tanggal 20 September 2014 sampai 25 September 2014, 20 Desember 2014 sampai 30 Desember 2014, serta 1 sampai 15 Januari 2015.

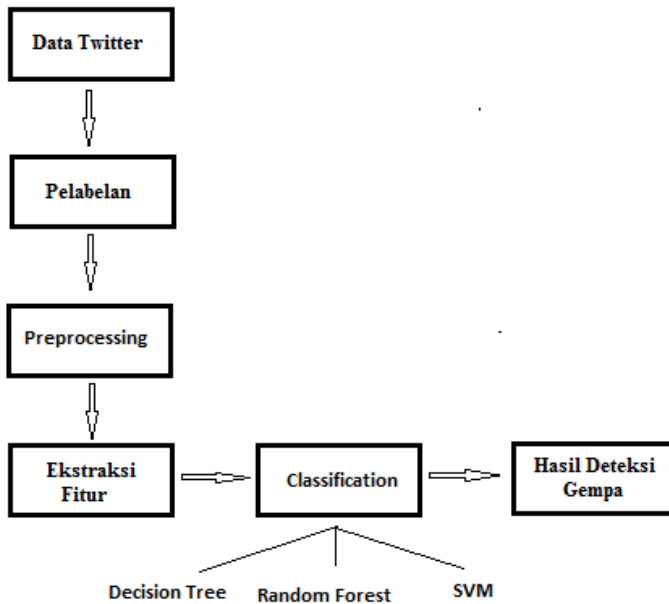
Perancangan Data Luaran

Data luaran yang didapatkan adalah hasil evaluasi akurasi model klasifikasi berdasarkan *cross-validation*, hasil evaluasi *TP Rate*, *FP Rate*, *precision*, *recall* dan *F-Measure* dalam aplikasi WEKA.

Sementara itu setelah didapatkan hasil evaluasi akurasi dari masing-masing metode, selanjutnya didapatkan juga luaran berupa model klasifikasi terbaik yang didapatkan dengan cara membandingkan hasil luaran.

3.2. Perancangan Sistem

Rancangan tugas akhir deteksi gempa berdasarkan data Twitter dapat dilihat pada Gambar 3.1.



Gambar 3.1 Arsitektur Umum Sistem

Pelabelan

Proses dimulai saat sejumlah data yang sudah didapat dari Twitter diberi label class gempa atau tidak gempa yang dilakukan secara manual.

Preprocessing

Data yang sudah diberi label akan dilakukan *preprocessing*. Pertama dilakukan tokenisasi yaitu proses pemisahan kata dari data tweet menjadi token-token/bagian-bagian tertentu. Tokenisasi didapatkan berdasarkan spasi, enter, tab, titik, koma, dsb seperti pada Tabel 3.1 :

Tabel 3.1 Tokenisasi

No	Dokumen	Tokenisasi
1	Barusan saja kota tarakan dilanda gempa	Barusan, saja, kota, tarakan, dilanda, gempa
2	Barusan sekitar jam 5 Wita ada gempa kah di bali	Barusan, sekitar, jam 5, wita, ada, gempa, kah, di, bali
3	barusan terasa gempa di cibubur, pusat gempa terjadi dimana ya	Barusan, terasa, gempa, di, cibubur, pusat, gempa, terjadi, dimana, ya
4	Barusan terasa gempa di wilayah banda aceh dan sekitar.	Barusan, terasa, gempa, di, wilayah, banda, aceh, dan, sekitar

Data yang telah melalui tahapan tokenisasi selanjutnya akan diproses untuk dilakukan penyaringan kata yang muncul dalam jumlah besar/umum atau kata yang tidak baku dan tidak memiliki makna(*stopword*) seperti pada Tabel 3.2 :

Tabel 3.2 Stopword Removal

No	Dokumen	Stopword Removal
1	Barusan, saja, kota, tarakan, dilanda, gempa	kota, tarakan, dilanda, gempa
2	Barusan, sekitar, jam 5, wita, ada, gempa, kah, di, bali	jam 5, gempa, bali
3	Barusan, terasa, gempa, di, cibubur, pusat, gempa, terjadi, dimana, ya	gempa, cibubur, pusat, gempa, terjadi
4	Barusan, terasa, gempa, di, wilayah, banda, aceh, dan, sekitar	Gempa, wilayah, banda, aceh, sekitar

Selanjutnya data akan diproses untuk menghilangkan angka seperti pada Tabel 3.3 :

Tabel 3.3 Hilangkan Angka

No	Dokumen	Hilangkan Angka
1	kota, tarakan, dilanda, gempa	kota, tarakan, dilanda, gempa
2	jam 5 , gempa, bali	gempa, bali
3	gempa, cibubur, pusat, gempa, terjadi	gempa, cibubur, pusat, gempa, terjadi
4	Gempa, wilayah, banda, aceh, sekitar	Gempa, wilayah, banda, aceh, sekitar

Selanjutnya data akan diproses dengan cara normalisasi data agar menghilangkan kata yang berulang seperti pada Gambar 3.4 :

Tabel 3.4 Normalisasi

No	Dokumen	Normalisasi
1	kota, tarakan, dilanda, gempa	kota, tarakan, dilanda, gempa
2	gempa, bali	gempa, bali
3	gempa , cibubur, pusat, gempa , terjadi	gempa, cibubur, pusat, terjadi
4	Gempa, wilayah, banda, aceh, sekitar	Gempa, wilayah, banda, aceh, sekitar

Dalam proses preprocessing juga akan dilakukan menghilangkan karakter-karakter seperti ("'+=!&?*^~#_") seperti pada Gambar 3.5 :

Tabel 3.5 Hilangkan Karakter

No	Dokumen	Hilangkan Karakter
1	Barusan gempa?	Barusan gempa
2	Barusan gempa. okebye-_-	Barusan gempa. Okebye
3	Barusan gempa...	Barusan gempa
4	barusan jogja gempa kan!	barusan jogja gempa kan

3.2.1. Ekstraksi Fitur

Data akan dilakukan pembobotan menggunakan *Term Frequency (TF)*. *Term Frequency (TF)* merupakan *frequency* kemunculan *term* atau kata dalam dokumen. Contoh seperti pada Tabel 3.1 :

Dokumen 1 : Saya belajar menghitung nilai tf.

Dokumen 2 : Tf merupakan frekuensi kemunculan term pada dokumen.

Tabel 3.6 Perhitungan Term Frequency

Term	Dokumen 1	Dokumen 2
Saya	1	0
frekuensi	0	1
Belajar	1	0
Hitung	1	0
Nilai	1	0
Tf	1	1
Pada	0	1
Term	0	1
Dokumen	0	1

Setelah proses selesai kemudian dilakukan eksport matriks dalam format .xls. Matriks yang dieksport memiliki ukuran $n \times m$, dimana n merupakan jumlah dari data dan m merupakan jumlah dari term.

3.2.2. Perbandingan Klasifikasi Decision Tree, Random Forest dan SVM

Pada tahap ini, dilakukan klasifikasi untuk mendapatkan *tweet* yang dianggap sebagai sinyal positif. Dalam kasus ini *tweet* yang dianggap sinyal positif adalah *tweet* yang dimaksudkan penggunaannya untuk memberikan informasi tentang gempa yang baru dialaminya. *Tweet* yang mengandung kata gempa dalam arti lain, dianggap sebagai sinyal negatif.

Data yang digunakan adalah data *tweet* yang mengandung kata gempa dari 20 September 2014 sampai 25 September 2014, 20 Desember 2014 sampai 30 Desember 2014, serta 1 sampai 15 Januari 2015.

Proses klasifikasi menggunakan aplikasi WEKA dengan metode *Decision Tree*, *Random Forest*, dan *SVM*. *cross-validation* yang digunakan adalah *cross-validation* 3, 5 dan 10. Setelah didapatkan hasil dari *cross-validation* selanjutnya akan dibandingkan hasil dari *TP rate*, *FP rate*, *precision*, *recall* dan *F-Measure* dari ketiga metode.

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Bab ini membahas tentang implementasi dari perancangan sistem. Bab ini berisi proses implementasi dari setiap kelas pada semua modul. Bahasa pemrograman yang digunakan adalah bahasa pemrograman PHP.

4.1. Lingkungan Implementasi

Dalam implementasi algoritma digunakan perangkat-perangkat sebagai berikut:

Perangkat Keras

Spesifikasi perangkat keras yang digunakan saat implementasi ditunjukkan pada tabel 4.1

Tabel 4.1 Spesifikasi perangkat keras

Perangkat	Prosesor	Memori
Samsung RV418	Intel Core i3	4 GB

Perangkat Lunak

Berikut perangkat lunak yang digunakan saat implementasi:

1. Windows 8.1
2. Xampp
3. Sublime Text
4. Notepad++
5. Excel
6. WEKA

4.2. Implementasi Modul

4.2.1. Implementasi Modul Conn.php

Modul ini berfungsi untuk membuat koneksi pada MySQL server dan memilih database mana yang akan diakses. Pada kode program di bawah ini ditunjukkan bahwa database yang diakses adalah “rendra”.

```

1. <?php
2.     //koneksi ke MySQL
3.     $tes=mysql_connect("localhost","root","");
4.     //koneksi ke database rendra.sql
5.     $tess=mysql_select_db("rendra");
6. ?>

```

Gambar 4.1 Kode program conn.php

4.2.2. Implementasi Modul Index.php

Modul ini berisi halaman antar muka pengguna dengan 2 macam tombol yaitu tombol Submit Token dan Export Matrix ke Excel. Submit Token di baris 68 berfungsi untuk tokenisasi. Tokenisasi adalah pemisahan tweet menjadi kata, pembuangan kata umum(stopword), dan menghilangkan kata yang mengandung angka. Sedangkan tombol Export Matrix ke Excel baris 75 berfungsi untuk mendownload hasil matrix dari modul export_matriks2.php

```

1. <html>
2.     <head>
3.         <title>Tes</title>
4.     </head>
5.
6.     <body>
7.         <div>
8.             <?php
9.                 include("conn.php");
10.
11.                 if(isset($_POST['input']))
12.                 {
13.                     $stopwords = array();
14.
15.                     $query_stop = mysql_query("select *
16. from stopword");
17.                     while($fetch_stop=mysql_fetch_array
($query_stop))
{

```



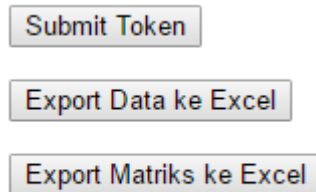
```

48.                                     if(preg_match_all("[
49.                                     {
50.                                         echo "Mengandung
51.                                         g Angka";
52.                                     }
53.                                     else
54.                                     {
55.                                         if(($jumlah_kar
56.                                             akter>3 && $jumlah_karakter<25))
57.                                             {
58.                                                 $input = my
59.                                                 sql_query("insert into term_twitter_class values('
60.                                                 , '$key', '$value', '$fetch_tweet[0]')");
61.                                                 if($input)
62.                                                 {
63.                                                     echo $y
64.                                                     ." ".$key." -> ".$value."<br>";
65.                                                 }
66.                                             }
67.                                         }
68.                                     }
69.                                     <form method="post" action="">
70.                                         <button name="input" value="input">
71.                                             Submit Token</button>
72.                                         </form>
73.                                     </div>
74.                                     <div>
75.                                         <p><a href="export_matriks2.php"><butto
76.                                             n>Export Matriks ke Excel</button></a></p>
77.                                         </div>
78.                                     </body>
79.                                     </html>

```

Gambar 4.2 Kode program index.php

Tampilan antarmuka pada Gambar 4.3:



Gambar 4.3 Halaman Antarmuka

4.2.3. Implementasi Modul Ganti_Kata_Class.php

Modul ini adalah modul yang bertugas untuk melakukan normalisasi kata. Normalisasi kata dilakukan untuk menghilangkan simbol seperti (,), [,], {, }, -, _, /, ~ pada baris 8. Selain itu juga dilakukan penghapusan pada term yang berulang pada baris 22.

```

1. <?php
2.     include("conn.php");
3.
4.     $query = mysql_query("select term, id_term_twitter_class, count(term) from term_twitter_class group by term");
5.     while($fetch_query=mysql_fetch_array($query))
6.     {
7.         $term = str_replace("(", "", $fetch_query[0]);
8.         $term = str_replace(")", "", $term);
9.         $term = str_replace("[", "", $term);
10.        $term = str_replace("]", "", $term);
11.        $term = str_replace("{", "", $term);
12.        $term = str_replace("}", "", $term);
13.        $term = str_replace("-", "", $term);
14.        $term = str_replace("_", "", $term);
15.        $term = str_replace("/", "", $term);
16.        $term = str_replace("~", "", $term);

```

```

17.
18.     $query_update = mysql_query("update term_twitter_class set term='$term' where id_term_twitter_class='$fetch_query[1]'");
19.     }
20.
21.     $query1 = mysql_query("select term, id_term_twitter_class, count(term) from term_twitter_class group by term");
22.     while($fetch_query1=mysql_fetch_array($query1))
23.     {
24.         if(preg_match_all("/[@]/", $fetch_query1[0]))
25.         {
26.             $query_hapus = mysql_query("delete from term_twitter_class where id_term_twitter_class='$fetch_query1[1]'");
27.         }
28.         elseif($fetch_query1[0]=="")
29.         {
30.             $query_hapus = mysql_query("delete from term_twitter_class where id_term_twitter_class='$fetch_query1[1]'");
31.         }
32.         elseif($fetch_query1[0]==" ")
33.         {
34.             $query_hapus = mysql_query("delete from term_twitter_class where id_term_twitter_class='$fetch_query1[1]'");
35.         }
36.         elseif(preg_match_all("/ /", $fetch_query1[0]))
37.         {
38.             $query_hapus = mysql_query("delete from term_twitter_class where id_term_twitter_class='$fetch_query1[1]'");
39.         }
40.     }
41.

```

```

42.     $query2 = mysql_query("select term, id_term_twitter_class, count(term) from term_twitter_class group by term");
43.     while($fetch_query2=mysql_fetch_array($query2))
44.     {
45.         echo "Hasil : ".$fetch_query2[0]."<br>";
46.     }
47. ?>

```

Gambar 4.4 Kode program export_matriks2.php

4.2.4. Implementasi Modul Export_Matriks2.php

Modul ini adalah modul yang bertugas untuk mengekspor matriks kedalam bentuk .xls. Matriks yang diexport berasal dari modul matriks2.php. Hasil export matriks disimpan dengan nama “matriks-export.xls” pada baris 6.

```

1. <?php
2.     // Fungsi header dengan mengirimkan raw data excel
3.     header("Content-type: application/vnd-ms-excel");
4.
5.     // Mendefinisikan nama file ekspor "hasil-export.xls"
6.     header("Content-Disposition: attachment; filename=matriks-export.xls");
7.
8.     // Tambahkan table
9.     include 'matriks2.php';
10. ?>

```

Gambar 4.5 Kode program evaluasi model di tweet_classifier

4.2.5. Implementasi Modul Matriks2.php

Modul ini berfungsi untuk melakukan pengambilan term dari table `term_twitter_class`. Selanjutnya untuk setiap term dilakukan penghitungan *Term Frequency(TF)*.

```

1. <table style="border:1;">
2.     <tr>
3.         <th>Dokumen</th>
4.     <?php
5.         include("conn.php");
6.         $dist_term = mysql_query("select
7.             t distinct(term) from term_twitter_class");
8.         while ($fetch_dist=mysql_fetch_
9.             array($dist_term))
10.            {
11.                ?>
12.                <th><?php echo $fetch_dist[0];
13.                ?></th>
14.                <?php
15.                }
16.                ?>
17.                <th>Class</th>
18.            </tr>
19.
20.            <?php
21.                $query1 = mysql_query("select *
22.                    from twitter_class");
23.                $query = mysql_query("select *
24.                    from twitter_class where class=0 limit 0,200");
25.                $jumlah_dokumen = mysql_num_row
26.                    s($query1);
27.                while($fq = mysql_fetch_array($
28.                    query))
29.                {
30.                    $query_term = mysql_query("
31.                        select distinct(term) from term_twitter_class");
32.                    ?>
33.                    <tr>
34.                        <td><?php echo $fq[0]; ?></
35.                        td>
36.                    <?php

```

```

29.         while($fetch_term = mysql_f
    fetch_array($query_term))
30.         {
31.             $query_df = mysql_query
    ("select distinct(id_twitter_class) from term_twitt
    er_class where term='$fetch_term[0]'");
32.             $query_jumlah = mysql_q
    uery("select count(a.jumlah) from term_twitter_clas
    s as a, twitter_class as b where b.kode='$fq[0]' an
    d a.term='$fetch_term[0]' and b.kode=a.id_twitter_c
    lass");
33.             $display_jumlah = mysql
    _fetch_array($query_jumlah);
34.             ?>
35.             <td><?php echo $display
    _jumlah[0]; ?></td>
36.             <?php
37.             }
38.             ?>
39.             <td><?php echo $fq[2]; ?></
    td>
40.         </tr>
41.     <?php
42.     }
43.     ?>
44. </table>

```

Gambar 4.6 Kode program matriks2.php

4.2.6. Tabel MySql

Modul yang sudah ada diimplementasikan menggunakan phpMyAdmin. Terdapat 5 tabel yang dipake, yaitu tabel stopwords, tabel twitter, tabel twitter_class, tabel term_twitter, dan tabel term_twitter_class seperti pada Gambar 4.7 :



Gambar 4.7 Tabel MySQL

Tabel stopwords

Tabel stopwords berisi kata umum atau yang tidak baku dengan jumlah 988 data. Tabel stopwords berisi field id_stopword sebagai primary key dan term_stopword seperti pada Gambar 4.8

		id_stopword	term_stopword
<input type="checkbox"/>	Ubah Salin Hapus	1	a
<input type="checkbox"/>	Ubah Salin Hapus	2	about
<input type="checkbox"/>	Ubah Salin Hapus	3	above
<input type="checkbox"/>	Ubah Salin Hapus	4	acara
<input type="checkbox"/>	Ubah Salin Hapus	5	across
<input type="checkbox"/>	Ubah Salin Hapus	6	ada
<input type="checkbox"/>	Ubah Salin Hapus	7	adalah

Gambar 4.8 Tabel Stopword

:

Tabel twitter

Tabel twitter berfungsi untuk menyimpan data twitter. Di tabel twitter berisi field id sebagai primary key dan text. Contoh pada Gambar 4.9 :

			id	text
<input type="checkbox"/>	Ubah	Salin	Hapus	1 gempa 5.0 sr baratdaya pacitan,jatim 25/9/2014 16 ...
<input type="checkbox"/>	Ubah	Salin	Hapus	2 mohon info gempa sore ini di wilayah semarang, bbr...
<input type="checkbox"/>	Ubah	Salin	Hapus	3 ya wajar lah banyak orang ngestatus gempa kan mmng...
<input type="checkbox"/>	Ubah	Salin	Hapus	4 ya tuhan aku kebangun gara2 gempa , ini ngerasain ...
<input type="checkbox"/>	Ubah	Salin	Hapus	5 4,6 sr di bali tidak berpotensi menimbulkan tsunam...
<input type="checkbox"/>	Ubah	Salin	Hapus	6 siwon merupakan anggota suju yang religious. terbu...

Gambar 4.9 Tabel Twitter

Tabel twitter_class

Tabel twitter_class berfungsi untuk memberikan label setiap isi dari twitter. Di dalam tabel twitter_class terdapat field kode, text, dan class. Field kode sebagai primary key, text berisi tweet, dan class berisi label. Label ada 2 macam yaitu 0 (bukan gempa) dan 1 (gempa). Contoh pada Gambar 4.10 :

kode	text	class
D1	- 25 September 2003 Gempa bumi berkekuatan 8.0 pad...	0
D10	rumah serasa muter- []hah gempa /	0
D100	aceh gempa mulu []	0
D1000	ngeri mental klo mentok masa RT /cengo/ @_imcongre...	0
D1001	ngga ahhhhh	0
D1002	ngga akh wkwk	0
D1003	ngga tau, tapi rasa rasanya kaya ada gempa gitu (0

Gambar 4.10 Tabel twitter_class

Tabel term_twitter

Tabel term_twitter berisi kata yang telah melalui tahapan preprocessing. Tabel term_twitter berisi id_twitter_class, term, jumlah, dan id_twitter. Id_twitter_class merupakan primary key. Contoh pada Gambar 4.11 :

id_term_twitter	term	jumlah	id_twitter
1	gempa	1	1
2	baratdaya	1	1
3	pacitan	1	1
4	jatim	1	1
5	kdllmn	1	1
6	dirasakan	1	1

Gambar 4.11 Tabel term_twitter

Tabel term_twitter_class

Tabel term_twitter_class berisi kata dari isi twitter yang terdapat pada tabel twitter_class. Tabel term_twitter_class ada 4 field seperti pada Gambar 4.12 :

id_term_twitter_class	term	jumlah	id_twitter_class
1	September	1	D1
2	Gempa	1	D1
3	bumi	1	D1
4	berkekuatan	1	D1
5	skala	1	D1

Gambar 4.12 Tabel term_twitter_class

4.3. WEKA

Proses Penggunaan Aplikasi WEKA bisa dilihat pada Gambar 4.6 – Gambar 4.11.

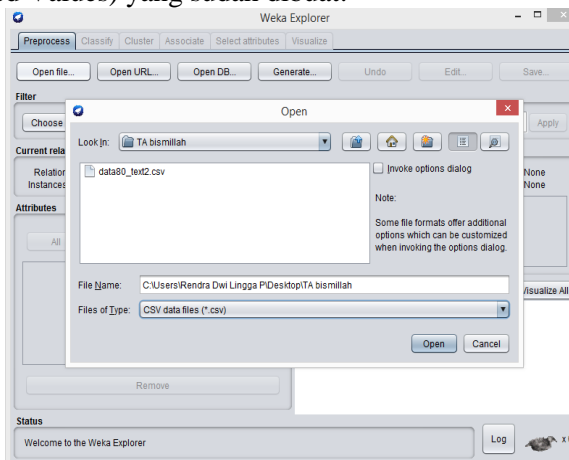
Pada Gambar 4.6 merupakan halaman awal aplikasi Weka, terdapat 5 tombol, tekan tombol Explorer yang digunakan untuk menggali lebih jauh data yang akan diproses

etela
h
mene
kan
tomb
ol
expl
rer,
pada
Gam
bar
4.7
akan
munc



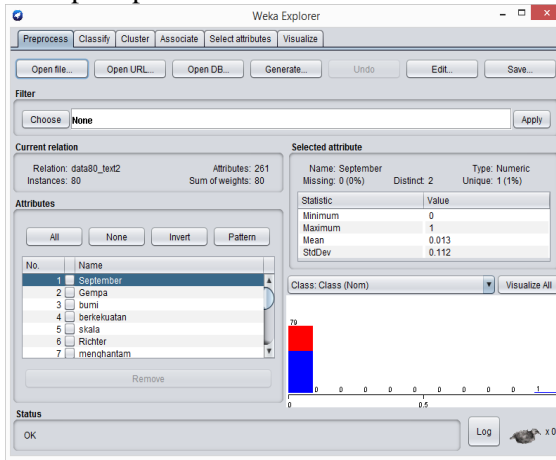
Gambar 4.13 Halaman Awal WEKA

ul halaman Weka Explorer. Selanjutnya buka file yang akan dieksekusi oleh Weka, penggunaan extensi file .csv (Command Separated Values) yang sudah dibuat.



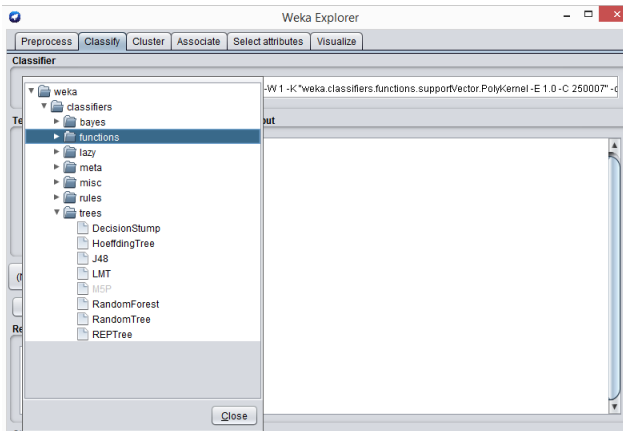
Gambar 4.14 Onen File

Setelah file terbuka maka akan muncul atribut-atribut dari data file .csv seperti pada Gambar 4.8 :



Gambar 4.15 Preprocess

Kemudian pilih algoritma yang akan digunakan seperti pada Gambar 4.9 :



Gambar 4.16 Pilihan Klasifikasi

D
alam
tugas
akhir
ini
meng
guna
kan
meto
de
klasif

ikasi algoritma *Decision Tree*, *Random Forest*, dan *SVM*.

Setelah dilakukan klasifikasi dengan masing-masing metode maka hasil yang didapat bisa dilihat pada Gambar 4.10 sampai Gambar 4.12 berupa nilai *FP Rate*, *TP rate*, *Precision*, *Recall*, dan *F-Measure*.

=== Detailed Accuracy By Class ===		Decision Tree			
	TP Rate	FP Rate	Precision	Recall	F-Measure
	0,960	0,533	0,750	0,960	0,842
	0,467	0,040	0,875	0,467	0,609
Weighted Avg.	0,775	0,348	0,797	0,775	0,755

Gambar 4.17 Klasifikasi Decision Tree

=== Detailed Accuracy By Class ===		Random Forest			
	TP Rate	FP Rate	Precision	Recall	F-Measure
	0,240	0,033	0,923	0,240	0,381
	0,967	0,760	0,433	0,967	0,598
Weighted Avg.	0,513	0,306	0,739	0,513	0,462

Gambar 4.19 Random Forest

=== Detailed Accuracy By Class ===		SVM			
	TP Rate	FP Rate	Precision	Recall	F-Measure
	0,980	0,300	0,845	0,980	0,907
	0,700	0,020	0,955	0,700	0,808
Weighted Avg.	0,875	0,195	0,886	0,875	0,870

Gambar 4.18 Klasifikasi SVM

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Pada bab ini, dibahas uji coba dan evaluasi dari implementasi metode-metode yang digunakan. Hal-hal yang diujikan dalam bab ini adalah: performa metode *Decision Tree*, *Random Forest* dan SVM untuk klasifikasi *tweet* gempa. Pengujian akan dilakukan dengan melihat nilai dari TP rate, FP rate, *Precision*, *Recall*, dan *F-Measure* pada masing-masing metode

5.1. Lingkungan Uji Coba

Lingkungan uji coba dalam tugas akhir ini meliputi perangkat lunak dan perangkat keras yang digunakan untuk implementasi metode. Perangkat keras yang digunakan adalah computer dengan prosesor Intel® Core™ i3-2310M dengan kecepatan 2.10GHz dan memori 4 gigabyte RAM. Uji coba dilakukan di sistem operasi Microsoft Windows 8.1 64bit dengan kakas bantu XAMPP dan WEKA

5.2. Data Uji Coba

Data uji coba yang digunakan dalam pengujian metode *Decision Tree*, *Random Forest*, dan SVM adalah data *tweet* yang mengandung kata gempa dari tanggal 20 September 2014 hingga 25 September 2014, 20 Desember 2014 sampai 30 Desember 2014, serta 1 Januari hingga 15 Januari 2015. Atribut dari *tweet* yang digunakan dalam uji coba ini adalah isi tulisan dari *tweet*.

5.3. Skenario Uji Coba

Di bagian ini dijelaskan skenario uji coba dalam proses pengujian. Ada 3 skenario uji coba, yaitu uji coba akurasi klasifikasi *Decision Tree*, akurasi klasifikasi *Random Forest*, dan akurasi klasifikasi SVM dari data Twitter.

5.3.1. Skenario Uji Coba Akurasi Klasifikasi Tweet dengan metode Decision Tree, Random Forest dan SVM

Pada skenario ini, digunakan 2278 *tweet* yang mengandung kata ‘gempa’ dari tanggal 20 September 2014 sampai 25 September 2014, 20 Desember 2014 sampai 30 Desember 2014, serta 1 sampai 15 Januari 2015. Dilakukan klasifikasi secara manual pada isi dari *tweet*, apakah *tweet* merupakan informasi dari seseorang tentang gempa yang baru dirasakan atau bukan.

Setelah dilakukan *preprocessing*, dilakukan klasifikasi dengan metode *Decision Tree*, *Random Forest*, dan *SVM* menggunakan aplikasi WEKA. Untuk proses klasifikasi akan dilakukan dengan 3-fold, 5-fold dan 10-fold *cross-validation*.

Data luaran yang didapat adalah *TP-Rate*, *FP-Rate*, *Precision*, *Recall*, dan *F-Measure* dengan melihat pada Tabel 5.1:

Tabel 5.1 Confusion Matrix

		Classified	
		Gempa	Bukan
Real	Gempa	TP	FP
	Bukan	FN	TN

TP Rate didapatkan dari persamaan 1:

$$TP\ Rate = TP / TP + FN \quad (1)$$

FP Rate didapatkan dari persamaan 2:

$$FP\ Rate = FP / FP + TN \quad (2)$$

Recall sama dengan persamaan 1:

$$Recall = TP\ Rate$$

Precision didapatkan dari persamaan 3:

$$Precision = TP / TP + FP \quad (3)$$

F-Measure didapatkan dari persamaan 4:

$$F-Measure = 2xPxR / P + R \quad (4)$$

Dengan TP sebagai jumlah gempa dirasakan yang terdeteksi sistem dan benar *real* gempa, TN sebagai jumlah gempa yang terdeteksi sistem dan bukan *real* gempa. FP sebagai jumlah bukan gempa yang terdeteksi sistem dan *real* gempa. FN sebagai jumlah bukan gempa yang terdeteksi sistem dan bukan *real* gempa. P adalah *Precision* dan R adalah *Recall*.

5.4. Hasil Uji Coba

Pada bagian ini akan dijelaskan hasil uji coba yang telah dilakukan.

5.4.1. Hasil Uji Coba Akurasi Metode Klasifikasi *Tweet* menggunakan Decision Tree

Hasil uji coba metode klasifikasi *tweet* dengan *Decision Tree* dilakukan dengan metode evaluasi 3-fold, 5-fold dan 10-fold *cross-validation*. Hasil dari evaluasi ditunjukkan pada Tabel 5.2:

Tabel 5.2 Hasil Uji Coba Akurasi Metode Klasifikasi Decision Tree 3-fold, 5-fold dan 10-fold Cross-Validation

n-fold Cross validation	Class Gempa			Class Bukan Gempa		
	3-fold	5-fold	10-fold	3-fold	5-fold	10-fold
TP Rate	46.7%	40.0%	36.7%	96.0%	96.0%	96.0%
FP Rate	04.0%	04.0%	04.0%	53.3%	60.0%	63.3%
Precision	87.5%	85.7%	84.6%	75.0%	72.7%	71.6%
Recall	46.7%	40.0%	36.7%	96.0%	96.0%	96.0%
F-Measure	60.9%	54.5%	51.2%	84.2%	82.8%	82.1%

Dari Tabel 5.2 adalah hasil uji coba metode *Decision Tree* dengan 3-fold, 5-fold, 10-fold *cross-validation* dalam mendeteksi gempa. *Recall* hasil dari *Decision Tree* 3-fold *cross-validation* dalam mendeteksi sebesar 46.7%. Namun *Recall* dalam mendeteksi bukan gempa lebih baik sebesar 96%.

5.4.2. Hasil Uji Coba Akurasi Metode Klasifikasi *Tweet* menggunakan *Random Forest*

Hasil uji coba metode klasifikasi *tweet* dengan *Random Forest* dilakukan dengan metode evaluasi 3-fold, 5-fold dan 10-fold *cross-validation*. Hasil dari evaluasi ditunjukkan pada Tabel 5.3:

Tabel 5.3 Hasil Uji Coba Akurasi Metode Klasifikasi *Random Forest* 3-fold, 5-fold dan 10-fold Cross-Validation

n-fold Cross validation	Class Gempa			Class Bukan Gempa		
	3-fold	5-fold	10-fold	3-fold	5-fold	10-fold
TP Rate	96.7%	96.7%	93.3%	24.0%	28.0%	34.0%
FP Rate	76.0%	72.0%	66.0%	03.3%	03.3%	06.7%
Precision	43.4%	44.6%	45.9%	92.3%	93.3%	89.5%
Recall	96.7%	96.7%	93.3%	24.0%	28.0%	34.0%
F-Measure	59.8%	61.1%	61.5%	38.1%	43.1%	49.3%

Dari Tabel 5.3 adalah hasil uji coba metode *Random Forest* dengan 3-fold, 5-fold, dan 10-fold *cross-validation* dalam mendeteksi gempa. *Recall* hasil dari *Random Forest* 3-fold *cross-validation* dalam mendeteksi gempa sebesar 96.7%.

5.4.3. Hasil Uji Coba Akurasi Metode Klasifikasi *Tweet* menggunakan SVM

Hasil uji coba metode klasifikasi *tweet* dengan SVM dilakukan dengan metode evaluasi 3-fold, 5-fold dan 10-fold *cross-validation*. Hasil dari evaluasi ditunjukkan pada Tabel 5.4:

Tabel 5.4 Hasil Uji Coba Akurasi Metode Klasifikasi SVM 3-fold, 5-fold dan 10-fold Cross-Validation

n-fold Cross validation	Class Gempa			Class Bukan Gempa		
	3-fold	5-fold	10-fold	3-fold	5-fold	10-fold
TP Rate	70.0%	63.3%	63.3%	98.0%	100%	100%
FP Rate	02.0%	0%	0%	30.0%	36.7%	36.7%
Precision	95.5%	100%	100%	84.5%	82.0%	82.0%
Recall	70.0%	63.3%	63.3%	98.0%	100%	100%
F-Measure	80.8%	77.6%	77.6%	90.7%	90.1%	90.1%

Dari Tabel 5.4 adalah hasil ujicoba metode SVM dengan 3-fold, 5-fold, dan 10-fold *cross-validation* dalam mendeteksi gempa. *Recall* hasil dari *SVM* 3-fold *cross-validation* dalam mendeteksi gempa sebesar 70%. Namun *Recall* dalam mendeteksi bukan gempa lebih baik sebesar 98%.

5.4.4. Hasil Perbandingan Persentasi Weighted Avg dari Decision Tree, Random Forest dan SVM

Hasil uji coba *weighted avg* dari metode *Decision Tree*, *Random Forest*, dan SVM. Menggunakan dengan metode evaluasi 3-fold, 5-fold dan 10-fold *cross-validation* ditunjukkan pada Tabel 5.5:

Tabel 5.5 Perbandingan *weighted avg*

3-fold Cross validation	Decision Tree	Random Forest	SVM
TP Rate	77.5%	51.3%	87.5%
FP Rate	34.8%	30.6%	19.5%
Precision	79.7%	73.9%	88.6%
Recall	77.5%	51.3%	87.5%
F-Measure	75.5%	46.2%	87.0%

5-fold Cross validation	Decision Tree	Random Forest	SVM
TP Rate	75.0%	53.8%	86.3%
FP Rate	39.0%	29.1%	22.9%
Precision	77.6%	75.1%	88.7%
Recall	75.0%	53.8%	86.3%
F-Measure	72.2%	49.8%	85.4%

10-fold Cross validation	Decision Tree	Random Forest	SVM
TP Rate	73.8%	56.3%	86.3%
FP Rate	41.1%	28.9%	22.9%
Precision	76.5%	73.1%	88.7%
Recall	73.8%	56.3%	86.3%
F-Measure	70.5%	53.9%	85.4%

Dari Tabel 5.5 dapat terlihat bahwa secara keseluruhan nilai *Recall* metode SVM dalam mendeteksi gempa menggunakan 3-fold *cross-validation* lebih baik daripada *Decision Tree* dan *Random Forest* dengan persentase 87.5%.

5.4.5. Hasil Uji Coba Perbandingan Persentase Recall yang Dideteksi oleh Sistem

Pada Tabel 5.6 adalah hasil dari uji coba yang dilakukan terhadap *recall* dari metode DT (*Decision Tree*), RF (*Random Forest*), dan SVM.

Tabel 5.6 Nilai *Recall* dari Sistem dalam Mendeteksi Gempa

3-fold cross validation	Gempa			Bukan Gempa		
	DT	RF	SVM	DT	RF	SVM
Recall	46.7%	96.7%	70.0%	96.0%	24.0%	98%

5-fold cross validation	Gempa			Bukan Gempa		
	DT	RF	SVM	DT	RF	SVM
Recall	40.0%	96.7%	63.3%	96.0%	28.0%	100%

10-fold cross validation	Gempa			Bukan Gempa		
	DT	RF	SVM	DT	RF	SVM
Recall	36.7%	93.3%	63.3%	96.0%	34.0%	100%

Dari Tabel 5.6 persentase *Recall* terlihat metode SVM dalam mendeteksi class bukan gempa lebih baik daripada *Decision Tree* dan *Random Forest* dengan nilai 98%. Namun *Random Forest* dalam mendeteksi class gempa lebih baik daripada *Decision Tree* dan SVM dengan nilai persentase tertinggi menggunakan 3-fold *cross-validation* sebesar 96.7%.

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan tugas akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang.

6.1. Kesimpulan

Dari hasil selama proses perancangan, implementasi, serta pengujian dapat diambil kesimpulan sebagai berikut:

1. Dalam melakukan klasifikasi *tweet* yang mengandung informasi gempa, metode *Random Forest* memiliki akurasi *Recall* sebesar 96.7%, lebih baik bila dibandingkan dengan *Decision Tree* dan SVM
2. Secara rata-rata dalam melakukan klasifikasi *tweet* yang mengandung informasi gempa atau tidak, metode SVM memiliki akurasi *Recall* sebesar 87.5%, lebih baik bila dibandingkan dengan *Decision Tree* dan *Random forest* yang masing-masing nilai rata-ratanya 77.5% dan 51.3%.

6.2. Saran

Berikut saran-saran untuk pengembangan dan perbaikan sistem di masa yang akan datang. Di antaranya adalah sebagai berikut:

1. Data yang didapatkan sekarang adalah data sampling yang diberikan oleh Twitter. Untuk mendapatkan semua *tweet* yang masuk ke Twitter, dibutuhkan akses ke Firehose API yang membutuhkan biaya. Kedepannya hal ini bisa dikembangkan lebih lanjut.
2. Dalam tugas klasifikasi, dapat diimplementasikan metode klasifikasi lain seperti *logistic regression*. Selain itu teks Twitter juga dapat diolah lebih lanjut dengan menggunakan *stemming*.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] Twitter, "What is Twitter | About," 2014. [Online]. Available: <https://about.twitter.com/what-is-twitter>.
- [2] Geoscience Australia, "What is an Earthquake?," 2013. [Online]. Available: <http://www.ga.gov.au/scientific-topics/hazards/earthquake/basics/what>.
- [3] Albaqir. Haidar M, "Aplikasi Deteksi Lokasi Gempa dan Peringatan Dini Berdasarkan Data Twitter menggunakan Random Forest dan Partikel Filter." 2015.
- [4] A. S. Nugroho, "Support Vector Machine, Teori dan Aplikasinya dalam Bioinformatika," IlmuKomputer.com, 2003.
- [5] L. Breiman, "Random Forests," *Machine Learning*, 2001.
- [6] S.Sayad, "Decision Tree Classification,". [Online]. Available: http://www.saedsayad.com/decision_tree.htm
- [7] T. Cheese, "Random Forest | Kaggle," 5 11 2014. [Online]. Available: <https://www.kaggle.com/wiki/RandomForests>.

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS

Penulis **Rendra Dwi Lingga P**, lahir di Lumajang pada 12 Oktober 1992. Penulis adalah anak ke-2 dari 3 bersaudara dan dibesarkan di Lumajang.

Penulis menempuh pendidikan formal di SDN Ditotrunan 1 Lumajang (1999-2005), SMPN 1 Sukodono Lumajang (2005-2008), dan SMAN 2 Lumajang (2008-2011). Pada tahun 2011, penulis menempuh pendidikan S1 jurusan Teknik Informatika Fakultas Teknologi Informasi di Institut Teknologi Sepuluh Nopember, Surabaya, Jawa Timur.

Di jurusan Teknik Informatika, penulis mengambil bidang minat KCV dan memiliki minat pada beberapa subjek seperti *machine learning* dan *data mining*.